Comparative Study of Expansion Functions for Evolutionary Hybrid Functional Link Artificial Neural Networks for Data Mining and Classification

Faissal MILI, Manel HAMDI

Applied Economics and Simulation,
Faculty of Management and Economic Sciences, 5100 Mahdia, Tunisia

International Finance Group Tunisia ,
Faculty of Management and Economic Sciences of Tunis, Tunisia

Abstract— This paper presents a comapraison between different expansion function for a specific structure of neural network as the functional link artificial neural network (FLANN). This technique has been employed for classification tasks of data mining. In fact, there are a few studies that used this tool for solving classification problems, and in the most case, the trigonometric expansion function is the most used. In this present research, we propose a hybrid FLANN (HFLANN) model, where the optimization process is performed using 3 known population based techniques such as genetic algorithms, particle swarm and differential evolution. This model will be empirically compared using different expansion function and the best function one will be selected..

Keywords-Expansion function; Data mining; Classification; Functional link artificial neural network; genetic algorithms; Particle swarm; Differential evolution.

I. Introduction

Classification task is a very important topic in data mining. A lot of research ([1], [2], [3]) has focused on the field over the last two decades. The Data mining is a knowledge discovery process from large databases. The extracted knowledge will be used by a human user for supporting a decision that is the ultimate goal of data mining. Therefore, classification decision is our aim in this study. A various classification models have been used in this regard. M. James [4] has employed a linear/quadratic discriminates techniques for solving classification problems. Another procedure has been applied using decision trees ([5], [6]). In the same context, Duda et al. [7] have proposed a discriminant analysis based on the Bayesian decision theory. Nevertheless, these traditional statistical models are built mainly on various linear assumptions that will be necessary satisfied. Otherwise, we cannot apply these techniques for classification tasks. To overcome the disadvantage, artificial intelligent tools have been emerged to solve data mining classification problems. For this purpose, genetic algorithms models were used [8]. In a recent research, Zhang ([9], [10]) have introduced the neural networks technique as a powerful classification tool. In these studies, he showed that neural network is a promising alternative tool compared to various conventional classification techniques. In a more recent literature, a specific structure of neural network has been employed for classification task of data mining as the functional link artificial neural network (FLANN). In fact, there are a few studies ([11], [12], [13]) used this tool for solving classification problems.

In this present research, we propose a hybrid FLANN (HFLANN) model based on three metaheuristics population based optimization tools such: genetic algorithms (GAs), particle swarm optimization (PSO) and differential evolution. This model will be compared using different expansion function and the best one will be selected.

`

II. Concepts and Definition

A. Population Based Algorithms

Population based algorithms are classed as a computational intelligence techniques representing a class of robust optimization ones. These population based ones make use of a population of solution in the same time based on natural evolution.

Many population based algorithms are presented in the literature such evolutionary programming [14], evolution strategy [15], genetic algorithms [16], genetic programming [17], Ant Colony [18], particle swarm [19] and differential evolution [20]. These algorithms differ in selection, offspring generation and replacement mechanisms. Genetic algorithms, particle swarm and differential evolutions represent the most popular ones.

1) Genetic algorithms

Genetic algorithms (GAs) are defined as a search technique that was inspired from Darwinian Theory. The idea is based on the theory of natural selection. We assume that there is a population composed with different characteristics. The stronger will be able to survive and they pass their characteristics to their offspring's.

The total process is described as follows:

1- Generate randomly an initial population;

2- Evaluate this population using the fitness function;

3- Apply genetic operators such selection, crossover and mutation;

4- Turn the process "Evaluation Crossover mutation" until reaching the stopped criteria fixed in prior.

2) Particle swarm

Presented in 1995 by L. Kennedy and R. Eberhart [19], particle swarm optimization (PSO) represents one of the most known population-based approaches, where particles change their positions with time. These particles fly around in a multidimensional search space, and each particle adjusts its position according to its own experience and the experience of their neighboring, making use of the best position encountered by itself and its neighbors. The direction of a particle is defined by the set of neighboring and its correspondent history of experience.

An individual particle i is composed of three vectors:

- Its position in the V-dimensional search space

X□ □□□⃗ = (X□□,X□□,..., X□□)□

- The best position that it has individually found

P□ □□⃗ = (P□□, P□□,..., P□□)□

- Its velocity V□ □□□⃗ = (V□□, V□□,..., V□□)

`

Particles were originally initialized in a uniform random manner throughout the search space; velocity is also randomly initialized.

These particles then move throughout the search space by a fairly simple set of update equations. The algorithm updates the entire swarm at each time step by updating the velocity and position of each particle in every dimension by the following rules:

. V□□ = χ ∗ (W ∗ V□□ + C ∗ ε□ (P□□ − X□□) + C ∗ ε□ □P□□ − X□□□) (1)

X□□ = X□□ + V□□ (2)

Where in the original equations:

C is a constant with the value of 2.0 ε□ and ε□ are independent random numbers uniquely generated at every update for each individual dimension (n = 1 to V).

P□□ is the best position found by the global population of particle.

P□□ is the best position found by any neighbor of the particle.

W: the weight

χ : the constriction factor.

3) Differential evolution

Proposed by Storn and Price in 1995 [20], differential evolution represents a new floating evolutionary algorithm using a special kind of differential operator. Easy implementation and negligible parameter tuning makes this algorithm quite popular.

Like any evolutionary algorithm, differential evolution starts with a population. Differential evolution is a small and simple mathematical model of a big and naturally complex process of evolution. So, it is easy and efficient.

Firstly, there are five DE strategies (or schemes) that were proposed by R. Storn and K. Price [20]:

• **Scheme DE/rand/1 :**
$\omega = x1 + F^* (x2 - x3)$ (3)

• **Scheme DE/rand/2 :**
$\omega = x5 + F * (x1 + x2 - x3 - x4)$ (4)

• **Scheme DE/best/1:**
$\omega = xbest + F * (x1 - x2)$ (5)

• **Scheme DE/best/2:**
$\omega = xbest + F * (x1 + x2 - x3 - x4)$ (6)

• **Scheme DE/rand-to best/1:**
$\omega = x+\lambda^* (xbest-x1)+F * (x2-x3)$ (7)

Later, two more strategies were introduced [21].

We present the trigonometric scheme defined by:

`

$\omega = (x1 + x2 + x3)/3 + (p2 - p1) * (x1 - x2)$
$+ (p3 - p2) * (x2 - x3) + (p1 - p3) * (x3 - x1)$ (8)
$pi = |f(xi)/ (f(x1) + f(x2) + f(x3))|$, $i = 1, 2, 3$; (9)

F define the constriction factor generally taken equal to 0.5

x define the selected element

x1, x2, x3, x4 and x5 represent random generated elements from the population.

Many others schemes can be found in the literature [20].

B. Functional Link Artificial Neural Networks

The FLANN architecture was originally proposed by Pao et al. [22]. The basic idea of this model is to apply an expansion function which increases the input vector dimensionality. We say that the hyper-planes generated provide greater discrimination capability in the input pattern space. By applying this expansion, we needn't the use of the hidden layer, making the learning algorithm simpler. Thus, compared to the MLP structure, this model has the advantage to have faster convergence rate and lesser computational cost.

The conventional nonlinear functional expansions which can be employed are trigonometric, power series, Chebyshev polynomials or Chebyshev Legendre polynomials type. R. Majhi et al. [23], shows that use of trigonometric expansion provides better prediction capability of the model. Hence, in the present case, we aim to validate the best expansion function for the proposed model.

Let each element of the input pattern before expansion be represented as X(i), $1 < i < I$ where each element x(i) is functionally expanded as Zn(i) , $1 < n < N$, where N = number of expanded points for each input element. In this study, we take N=5.

I= the total number of features

As presented in figure 1, the expansion of each input pattern is done as follows.

$Z1(i) = X(i), Z2(i) = f1(X(i)),\ldots, Z5(i) = f5(X(i))$ (10)

These expanded inputs are then fed to the single layer neural network and the network is trained to obtain the desired output. Different expansion function will be described next.

C. Expansion function

Four expansion function will be used in this work such, trigonometric, the polynomial, the Legendre polynomial and the power series. Different characteristics are presented in the 4 graphics.

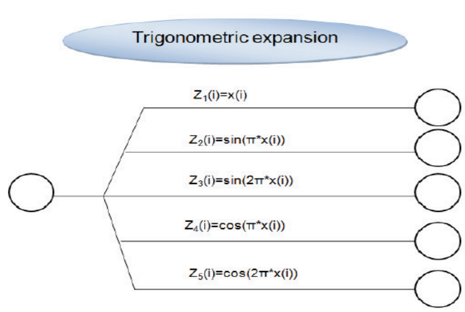


Figure 1. Trigonometric functional expansion of the first element

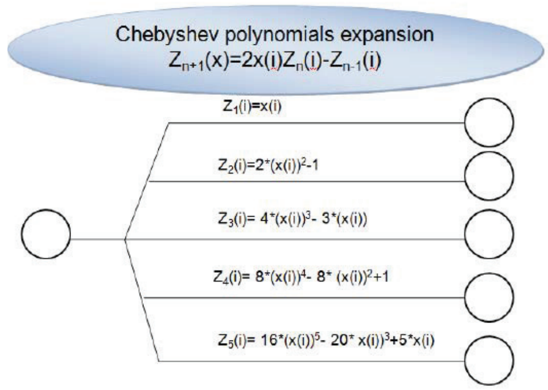


Figure 2. Chebyshev polynomials functional expansion of the first element

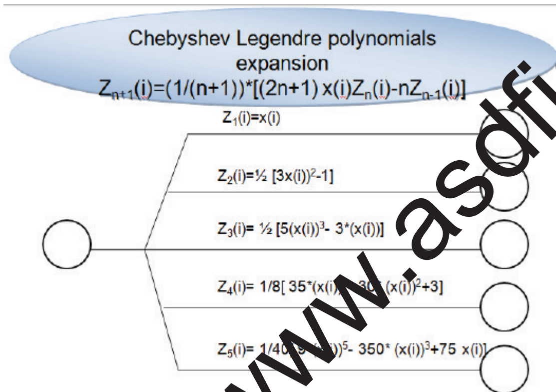


Figure 3. Chebyshev Legendre polynomials functional expansion of the first element

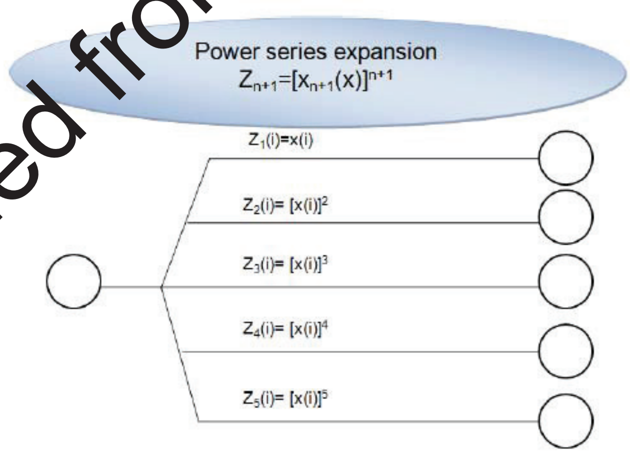


Figure 4. Power series functional expansion of the first element

III. Hybrid Flann Description

The proposed hybrid FLANN is based on evolutionary algorithms as genetic algorithms, particle swarm and differential evolution.

A. Resampling Technique*:*

In order to avoid over fitting, we use the (2*5) K fold cross validation resampling technique. We proceed as follows:

We divide initial database into 5 folds (K=5) where each one contain the same repartition of classes. For example, if initial population contains 60% of class 1 and 40% of class 2, then all the resulted K folds must have the same repartition.

B. Generation

We begin the process by generating randomly initial solution. We execute partial training using differential evolution in order to improve initial state.

C. Fitness Fuction and Evaluation

In order to evaluate each solution, two criterions are used such the mean square error (MSE) and the misclassification error (MCE) rate. If we have to compare solutions A and B, we apply the following rules: A is preferred to B If and only if MCE(A)< MCE(B) Or MCE(A)= MCE(B) and MSE(A)< MSE(B).

D. Selection

Many selections are defined in the literature such the Roulette wheel method, the N/2 elitist method and the tournament selection method. The last method will be used here. The principle is to compare two solutions, and the best one will be selected.

N/2 elitist is used at the beginning of the process in order to select 50% of generated solution.

E. Crossover

Two parents are selected randomly in order to exchange their information. Two crossovers are applied and described as follows:

1) Crossover 1 (over input feature): An input feature is chosen randomly to exchange his correspondent weight between the selected two parents.

2) Crossover 2 (over output nodes): An output is chosen randomly to exchange his correspondent weight.

3) Crossover 3 (Crossover over connection): A connection position is chosen randomly and his correspondent weight is exchanged between the two parents.

F. Mutation

1) Mutation 1(over connection)

A connection position is chosen randomly and his correspondent weight has been controlled. If this connection is connected, his correspondent weight is disconnected by setting his value equal to zero. Else, this connection is connected.

2) Mutation 2 (over one input feature)

An input feature is chosen randomly and his correspondent weights have been controlled. If this input feature is connected (there is at least one weights of his correspondent ones is different from zero), it will

be disconnected by putting all his entire weight equal to zero. Else if this input feature is totally disconnected, it will be connected there by generating weights different from zero.

3) Mutation 3 (over two input feature)

We do the same like mutation 2 but here simultaneously for the two selected features.

4) Mutation 4 (over three input feature)

In this mutation, the same principle is used for three input features.

We note that many input features connection and disconnection can be executed in the same time when having a large number of features. This crossover helps to remove undesirable features from our classification process and can improve the final performance process.

G. Particle swarm optimization (PSO)

In the presented paper, we define three PSO model based on the notion of neighbor.
1) PSO based on resulted genetic offspring's: First, we apply genetic operators. Each offspring that improve our fitness function define a neighbor, and used in equation (1).

2) PSO based on Euclidian distance: For each particle, we compute the Euclidian distance between this particle and the rest of the population. Next we choose the five nearest particles based on this distance. From the selected subset of neighbors, we choose the best

one which has the best fitness value. This selected one defines our neighbor to be replaced in equation (1).

3) PSO based on the last best visited solution: In this case, each particle flies and memorizes his best reached solution. This memory defines the neighbor to be used in equation (1).

H. Differential evolution

In this work, we proceed as follows:

- First, for each candidate x, we generate five random solution x1, x2, x3, x4 and x5.
- Next we apply seven chosen schemes as follows:

DE1: Scheme DE/direct/1 :
$\omega = x + F^* (x2 - x1)$ (11)

DE2: Scheme DE/best/1 :
$\omega = xbest + F^* (x2 - x1)$ (12)

DE3: Scheme DE/best/1 :
$\omega = xbest + F^* (x3 - x2)$ (13)

DE4: Scheme DE/best/1 :
$\omega = xbest + F^* (x3 - x1)$ (14)

DE5: Scheme DE/best/2 :
$\omega = xbest + F * (x1 + x2 - x3 - x4)$ (15)

DE6: Scheme DE/rand/2 :
$\omega = x5 + F * (x1 + x2 - x3 - x4)$ (16)

`

DE7: with Trigonometric Mutation:

$$\omega = (x1 + x2 + x3)/3 + (p2 - p1) * (x1 - x2) + (p3 - p2) * (x2 - x3) + (p1 - p3) * (x3 - x1) \quad (17)$$

$$pi = |f(xi)/ (f(x1) + f(x2) + f(x3))|, i= 1, 2, 3 ; \quad (18)$$

I. Stopping criterion:

The process turns in a cycle until reaching a maximum number of epochs without any improvement. We fix the maximum number of epochs equal to 30 epochs.

IV. Experimental Studies:

11 real-world databases were selected there to be used in simulation works. They are chosen from the UCI repository machine learning, which is commonly used to benchmark learning algorithms [24].

We compare the results of the proposed hybrid FLANN (HFLANN) with FLANN based on the gradient descent algorithm. Next, Comparison with other classifiers will be done.

A. Description of The Databases

A brief description of used databases for experimental setup is presented in table I. Num. is the numeric features, Bin. is the binary ones, and Nom. is the nominal inputs that mean discrete with three or more distinct labels.

Table I. Summary of The Dataset Used in Simulation Studies

Dataset	Inputs				Ex.	Cls
	Num.	Bin.	Nom.	Total		
IRIS	4	0	0	4	150	3
VOTING	0	16	0	16	435	2
BREAST	0	0	9	9	699	2
PRIMA	8	0	0	8	768	2
CREDIT	6	4	4	14	690	2
BALANCE	4	0	0	4	625	3
WINE	13	0	0	13	178	3
BUPA	6	0	0	6	345	2
ECOLI	7	0	0	7	336	8
GLASS	10	0	0	10	214	6
ZOO	1	15	0	16	101	7

B. Initial population improvement:

A random generated population will be generated randomly and their performance is presented in column 2 of table II. Random generation gives worst results needing some initial improvement. For this aim, we propose to use two prior improving algorithms: the back-propagation, the differential evolution, and a mixed back-propagation differential evolution one. From column 3 and column 4, we observe that the back propagation one has a better improving performance than the differential evolution. In the last column,

mixed back propagation- differential evolution results are presented. Compared to single algorithm results, the mixed algorithm gives the better result and it we be used in our process as a prior improving algorithm.

Table II. Summary of The Dataset Used in Simulation Studies

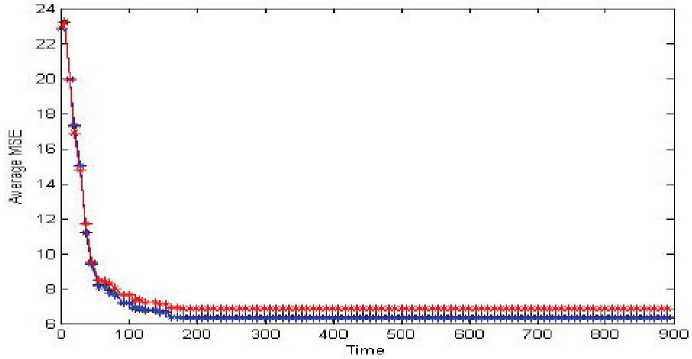
	Random Generation	Random Generation with BP	Random Generation with DE	Mixed BP DE
1	20,57096028	6,6667	23,3333	0
2	41,6667	0	41,6667	3,3333
3	65	0	36,6667	1,6667
4	65	0	48,3333	3,3333
5	38,3333	6,6667	21,6667	0
6	68,3333	6,6667	25	1,6667
7	58,3333	6,6667	25	0
8	83,3333	0	26,6667	0
9	76,6667	6,6667	16,6667	1,6667
10	96,6667	0	25	3,3333
11	45	0	25	0
12	38,3333	6,6667	30	1,6667
13	30	6,6667	30	1,6667
14	40	6,6667	21,6667	0
15	25	6,6667	38,3333	0
16	50	0	33,3333	0
17	35	0	41,6667	0
18	26,6667	6,6667	30	3,3333
19	25	6,6667	26,6667	0
20	40	0	13,3333	0
Mean	47,44521301	3,666685	29,000005	1,083335
Ecart Type	20,99774004	3,462802251	8,8921693	1,354538289

C. Convergence test:

In order to test the convergence of the proposed hybrid FLANN, a comparison will be done with trained FLANN using the back-propagation algorithm. Results are presented in figure 5 and figure 6. Comparison is done based on the required time and number of epochs for convergence.

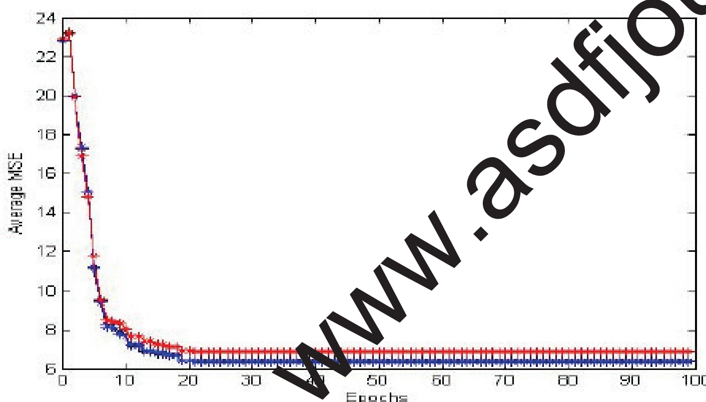
From figure 5, we find that our process needs less than 200 seconds 20 epochs to converge. Figure 6 present results for FLANN based on back-propagation. This model requires less than 150 seconds and 15 epochs to converge.

The proposed hybrid FLANN has a strong ability to converge fast and requires approximately the same time and epochs than FLANN based back-propagation.



a.MSE vs Time

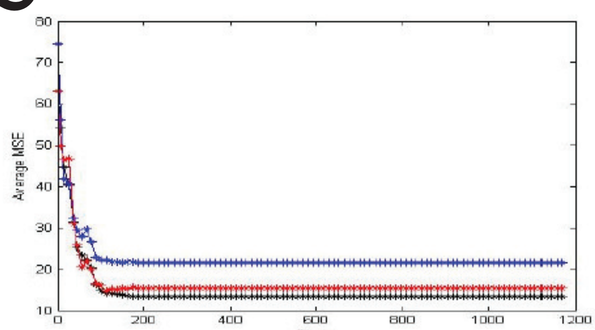
A.MSE vs Time



b.MSE vs epochs

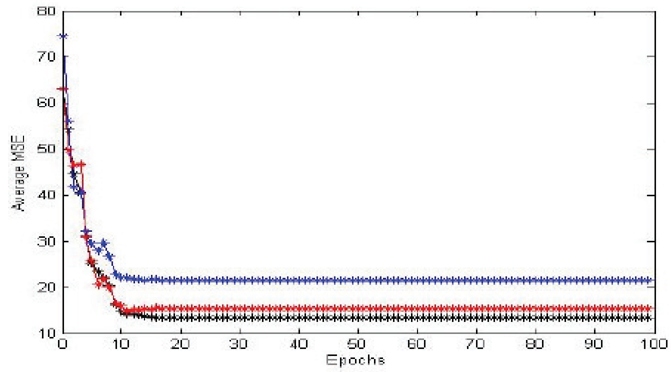
B.MSE vs epochs

Figure 5. The MSE Hybrid FLANN results vs. time and epochs applied to the iris database



a. MSE vs Time

A. MSE vs Time



b. MSE vs epochs

B. MSE vs epochs

Figure 6. The MSE FLANN based back-propagation results vs. time and epochs applied to the iris database

D. Comparative study:

Table III. Average Comparative Performance of Hflann Based Different Expansion Function

Database	FLANN Based BP	Trigonometric Hybrid FLANN with Local Search	Chebytchev Hybrid FLANN with Local Search	Power Series Hybrid FLANN with Local Search	Legend Chebytchev Hybrid FLANN with Local Search
IRIS	0,8933	0,96667	0,94667	0,94667	0,95333
VOTING	0,7829	0,94498	0,94049	0,94001	0,95877
BREAST	0,9298	0,9599	0,96704	0,96575	0,94259
PRIMA	0,6536	0,73964	0,73332	0,738	0,75142
CREDIT	0,5935	0,85316	0,83698	0,67253	0,84962
BALANCE	0,6036	0,89779	0,88669	0,86359	0,83266
WINE	0,9035	0,96111	0,92582	0,97222	0,90905
BUPA	0,5392	0,69328	0,66101	0,69849	0,62941
ECOLI	0,6279	0,81661	0,77389	0,8219	0,68185
GLASS	0,3463	0,61769	0,61047	0,53167	0,46039
ZOO	0,4163	0,85606	0,85126	0,88404	0,78934
Mean of Means	0,66272	0,84608	0,83033	0,82135	0,79619
Mean of Standard Deviation		0,05276	0,05419	0,06811	0,06124
Range		1	3	4	2

Table III present results of the proposed model using four different expansion functions. We find that our model gives better results that the FLANN based back-propagation algorithms.

By comparing different expansion function, we find that the trigonometric expansion function is the best one having the best mean of performance (0.84608), and the little mean of standard deviation (0.05276). This expansion function gives the best results over 7 databases from 11.

We can conclude that the trigonometric expansion function is the best one.

`

V. Conclusion

A HFLANN was proposed based on three populations based algorithms such genetic algorithms, differential evolution and particle swarm. This classifier shows his ability to converge faster and gives better performance than FLANN based on back-propagation.

Based on our experimentation, and compared to others expansion function, the trigonometric one is found the best one.

In future work, we can add a wrapper approach able to delete automatically irrelevant features. We can also apply the HEFLANN to others data mining problems such prediction.

Others evolutionary algorithms can be included in the proposed process in order to perform better results. Others comparison criteria can be used such the needed speed and the robustness of the algorithm

References

[1] R .Agrawal, T. Imielinski and A. Swami, “Database mining: A performance perspective”. IEEE Trans. Knowledge Data Eng., 5, pp. 914-925 (1993).

[2] U. M. Fayyad, Piatetsky-Shapiro, G., and Smyth, P., “From data mining to knowledge discovery: An overview”. In U. M. Fayyad, G. Piatetsky- Shapiro, & P. Smyth (Eds.), Advances in knowledge discovery and data mining ,pp. 1–34,. Menlo Park, CA: AAAI Press, 1996.

[3] H.P. Kriegel, et al., “Future trends in data mining”. Data Mining and Knowledge Discovery, 15(1), 87–97. Netherlands: Springer (2007).

[4] M., James, “Classification Algorithms”. Wiley,1985.

[5] L. Breiman, , J.H. Friedman, R.A. Olshen and C.J. Stone, “Classification and Regression Learning”. Morgan Kaufman, 1984.

[6] H.P. Kriegel, et al., “Future trends in data mining”. Data Mining and Knowledge Discovery, 15(1), 87–97. Netherlands: Springer (2007).

[7] R.O. Duda, Hart PE and Stork D.G., “Pattern classification”. Wiley, New York ,2001.

[8] D.E. Goldberg, “Genetic algorithms in search, optimization and machine learning”. Morgan Kaufmann ,1989.

[9] G. P., Zhang, “Neural networks for classification”, A survey. IEEE Transactions on Systems, Man, Cybernetics Part C: Application and Reviews, 30(4), pp. 451–461, 2000.

[10] G. P Zhang, “Avoiding pitfalls in neural network research”. IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, 37(1), pp. 3–16 , 2007. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[11] R. Majhi, G. Panda and G. Sahoo, “Development and performance evaluation of FLANN based model for forecasting of stock markets”, Expert Systems with Applications 36, pp. 6800–6808,2009.

[12] S. Dehuri and Cho S.B. ,”A hybrid genetic based functional link artificial neural network with a statistical comparison of classifiers over multiple datasets”. Neural Comput & Applic (19), pp.317-328. ,2010a.

[13] S. Dehuri and Cho S.B.,"Evolutionarily optimized features in functional link neural network for classification". Expert Systems with Applications (37), pp.4379–4391, 2010b .

[14] L., Fogel, J. Owens and J. Walsh, "Artificial Intelligence through Simulated Evolution". John Wiley, Chichester,1966.

[15] D.E. Goldberg, "Genetic algorithms in search, optimization and machine learning". Morgan Kaufmann ,1989.

[16] J., Holland, "Adaptation in natural and artificial systems". Univ. of Michigan Press, Ann Arbor ,1975.

[17] J. Koza, , "Genetic programming on the programming of computers by means of natural selection", Cambridge MA: MIT Press, 1992.

[18] M. Dorigo, , Stutzle, T.,"Ant Colony Optimization". MIT Press, Cambridge. ,2004.

[19] J., Kennedy Eberhart R., "Particle Swarm Optimization", In Proceedings of IEEE International Conference on Neural Networks, pp. 1942– 1948,1995.

[20] R. Storn Kenneth P. "Differential evolution A simple and efficient adaptive scheme for global optimization over continuous spaces". Technical Report TR, pp. 95-012, International Computer Science Institute, Berkeley, CA, 1995.

[21] F. Hui-Y. and J. Lampinen,"A trigonometric mutation approach to differential evolution". In K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, & T. Fogarty, editors, Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pp. 65–70, Athens, Greece. International Center for Numerical Methods in Engineering (Cmine) , 2001.

[22] Y.-H. Pao, S.M. Phillips and D.J. Sobajic, "Neural-net computing and intelligent control systems". Int. J. Contr., 56, pp. 263-289 , 1992.

[23] R. Majhi, G. Panda and G. Sahoo, "Development and performance evaluation of FLANN based model for forecasting of stock markets", Expert Systems with Applications 36, pp. 6800–6808,2009.

[24] C.L. Blake,. and Merz, C.J., 'UCI Repository of machine learning databases', em Irvine, CA: University of California, department of information and Computer Science. Available on-line at : http//www.ics.uci.edu/~mlearn/ML.Repository.html. , 1998.

`